Mapping Attribution Metadata to the Open Provenance Model

Simon Miles[a]

[a]*Department of Computer Science, King's College London, London, WC2R 2LS, UK*

Abstract

A description of a data item's provenance can be provided in different forms, and which form is best depends on the intended use of that description. Because of this, different communities have made quite distinct underlying assumptions in their models for electronically representing provenance. Approaches deriving from the library and archiving communities emphasise agreed vocabulary by which resources can be described and, in particular, assert their *attribution* (who created the resource, who modified it, where it was stored etc.) The primary purpose here is to provide intuitive metadata by which users can search for and index resources. In comparison, models for representing the results of scientific workflows have been developed with the assumption that each event or piece of intermediary data in a process' execution can and should be documented, to give a full account of the experiment undertaken. These occurrences are connected together by stating where one derived from, triggered, or otherwise caused another, and so form a *causal graph*. Mapping between the two approaches would be beneficial in integrating systems and exploiting the strengths of each. In this paper, we specify such a mapping between Dublin Core and the Open Provenance Model. We further explain the technical issues to overcome and the rationale behind the approach, to allow the same method to apply in mapping similar schemes.

Key words: provenance, OPM, attribution, e-science, Dublin Core

1. Introduction

Provenance, i.e. something's history or source, is a critical topic in many different domains and, because of this, the electronic representations of provenance used can vary in their conceptual underpinnings. Such variation makes it harder to avoid repetition or provide interoperation where there is integration of previously independent systems. In particular, some describe the provenance of a resource in terms of *attribution metadata*, stating who created or modified it, and where and when, while others model provenance as a *causal graph*, in which

Email address: simon.miles@kcl.ac.uk (Simon Miles)

occurrences trigger or influence each other, in the end leading up to the resource being as it is (the resource's *lineage*). It is the aim of this paper to present a broad approach to mapping between the two, with a focus on integrating data from two models: Dublin Core (DC) [1] and the Open Provenance Model [2].

The core of this paper is a specification of how provenance-related DC metadata terms map to patterns in OPM graphs, and vice-versa. The intention is to allow existing Dublin Core or Open Provenance Model data to be re-expressed in the other model. Specifically, the motivating goals are as follows: *(i)* to allow currently existing provenance-related metadata expressed using DC to be exported as an OPM graph, so that services capable of parsing OPM can query that graph and integrate with OPM data on connected processes; *(ii)* to allow currently existing data expressed using the Open Provenance Model to be exported as a DC RDF graph, so that services capable of parsing DC RDF can query and search across that data; *(iii)* to provide a way for those currently using DC to start adding metadata regarding the processes which produced their resources, e.g. to specify exactly *how* a resource came to be created, the order in which contributions were made to a resource, describe what is shared in the history of two resources without repetition etc.

The work described here derives from preliminary efforts on an OPM profile specification to map between DC and OPM data and it is hoped that feedback from readers of this paper will be used to improve the profile to the point at which it can become widely accepted and officially endorsed. However, this paper is not itself the specification, and its scope aims to be wider by providing discussion on the technical challenges of mapping between the two viewpoints.

2. Mapped Technologies

In this section, we introduce each technology used in the mapping, and discuss their strengths and weaknesses. Dublin Core is an example of an approach to expressing attribution metadata, i.e. assertions that a user or organisation played a particular role in the life of the resource or that, in such a role being played, the resource was affected on some date or by some method. The Open Provenance Model uses causal graphs to describe the lineage of processes leading up to a data artifact being as it is.

2.1. Dublin Core

Dublin Core provides a vocabulary for describing *resources*, and its strength comes from shared usage across disparate repositories and organisations. By using a common vocabulary it is, for example, possible to search or index across distributed resources, and remote applications can use DC terms in communication about resources. It emerged from the library and archiving communities, but is prevalent in Web-focused research, allowing the annotation of web-accessible and other resources with agreed metadata [3].

Dublin Core consists of a core set of metadata elements and a set of qualifiers which make it clearer how to interpret the elements. Of these a subset of

elements and qualifiers can be seen as regarding provenance, particularly in the sense of attribution. For example, there are terms for the `creator` of a resource, for its `publisher`, and the `date` of its publication.

A typical serialisation of a DC metadata assertion consists of: *(i)* an identifier for the resource being described (sometimes implied by the context of the assertion); *(ii)* a term from the DC vocabulary, qualified or not; and, *(iii)* the annotation value. For example, "this paper" has a "title" of "Mapping Attribution Metadata to the Open Provenance Model". This information can be encoded in many data formats, but the Resource Description Framework (RDF) fits the form of information (i.e. triples) well, and DC provides URI versions for each of its terms, to be used as RDF properties. In the RDF serialisation, qualified terms are sub-properties of the unqualified terms. We will take the RDF realisation of DC as the main starting point for describing our mapping.

2.2. Open Provenance Model

The Open Provenance Model (OPM) is a representation of the processes which have led to data being produced or transformed into a new state, and so can represent the provenance of one or more data items. Here we will summarise only the part of OPM we need for this paper, and follow the OPM v1.1 specification. The reader is encouraged to read the specification for more detail.

OPM is a causal graph model of provenance, meaning that an OPM description of provenance is a graph whose edges denote causal relationships (X was caused by Y) between the occurrences denoted by the nodes. This structure allows OPM graphs to describe how multiple events led to some data being produced (serially or independently), how one piece of data was derived from another, etc. OPM classifies occurrences (nodes) into three types: *artifacts*, *processes* and *agents*. Artifacts are pieces of data of fixed value and context, possibly representing an entity in a given state; processes are (non-instantaneous) actions which are performed on artifacts to produce other artifacts; and agents denote the entities controlling process execution, such as users. The properties which artifacts, processes and agents possess can be documented by arbitrary key-value *annotations* to the nodes. Edges can also have annotations to provide further information on *how* one occurrence caused another.

2.3. Strengths and Weaknesses

Attribution metadata, such as DC, is intuitive for describing resources, as it is based on a simple set of terms corresponding to those used for searching for journals, books and other artifacts in repositories. As such it is a good language for user-computer interaction regarding metadata, and the terms have been used to develop forms in user interfaces.

DC does not aim to have an unambiguous semantics, in reflection of the range of its applications [4]. Qualified terms aim to convey more precise information than unqualified terms, but still encompass multiple interpretations. For this reason, a one-to-one mapping, independent of application, to another representation is implausible. Instead, we aim for a base mapping to OPM for simple interpretations of terms, which can be refined on by specific applications.

OPM graphs include structured information which is at least non-trivial to represent using computer-parsable attribution metadata. For example, it is a simple matter in OPM to provide increasingly detailed accounts of how a user contributed to a document, rather than just stating the fact of the contribution alone. This is because there is an explicit and general model of past processes and the entities involved in them.

The viewpoint of OPM is different from that of DC. OPM deals with fixed, well-defined past occurrences, while DC describes resources and users that may change over time, and can describe the present state as well as the past. In particular, an OPM artifact is not the same as a DC resource: an artifact represents a resource in one particular state and context.

3. Integration of OPM and Dublin Core

We propose a mapping between provenance information in Dublin Core and the Open Provenance Model. In mapping, we faced several issues: *(i)* DC refers to data as resources which may change over time, whereas OPM only models data artifacts at fixed instants; *(ii)* a single DC assertion generally does not correspond to a single causal relationship in OPM, so we have to map assertions to *patterns* in OPM graphs; *(iii)* some DC terms can refer to what occurred in the past or to the present, so do not map to provenance data in all cases (others do not describe provenance at all). We use identifiers from multiple specifications in the mapping, so use distinguishing namespace prefixes: `dc` for DC, `opm` for OPM, and `map` for the DC-OPM mapping specification.

3.1. Mapping Mutable to Immutable Data

DC may refer to *mutable* resources, such as users or documents. Multiple artifacts in an OPM graph can represent different versions or points within a mutable resource's lifetime, but no graph node can represent the resource itself. Mapping from attribution metadata to causal graphs requires retaining information that multiple nodes denote instances of one mutable resource.

Before we describe the mapping itself, it is important to clarify what it means for an immutable artifact to be related to a mutable resource. One way to describe it would be to say that each artifact is a different 'version' of the resource. Unfortunately, DC uses 'version' to imply that the resource has changed value from one instance to the other, while in OPM two artifacts may be distinguished if they differ in context alone, e.g. a file on my local filesystem is not the same artifact as the same data uploaded to a database, even though the data content is the same. Therefore, following the Metadata Encoding and Transmission Standard (METS) [5], we use the term 'generation' to avoid the connotations of 'version' and other such commonly used terms.

We propose two ways for mutation information to be included in an OPM graph, and each can be helpful for different purposes. First, we can augment a `opm:wasDerivedFrom` relation to state that, not only was one artifact derived from another, but they are both generations of the same resource (without naming that resource). To allow this, we define the annotation value

`map:laterGenerationThan` to sub-type the edge from the later generation of the resource to the earlier. Second, we can name the resource for which an artifact is a generation. We include this name as an annotation to the artifact of key `map:isGenerationOf` and whose value is the identifier of the resource. We will use both approaches in combination in the mapping below.

3.2. Mappings

Mappings between relevant DC terms and OPM graphs are given below. Each is based around an OPM graph in which some annotation values have been replaced by unbound variables, so making it a *pattern* for graphs documenting processes of the same form. For each pattern, we specify the DC terms which map to it, under what definition they apply, the OPM pattern as a graph, a general description, and a specific example. In some cases, we also give 'constraints', which are factors other than the given graph structure and fixed annotation values which dictate whether an OPM graph maps to a given DC assertion.

In using a mapping, a single DC assertion matching one of those listed in a pattern would be translated into the OPM graph, or an OPM graph fragment matching that shown in a pattern would translate to the appropriate DC assertion. Translating multiple DC assertions would result in multiple disconnected graphs. Additional information, e.g. knowing that a contribution immediately preceded publication of a resource, could then allow the graphs to be unified by treating artifacts in two graphs as denoting a single state of a resource.

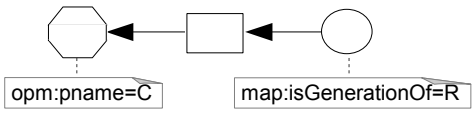
The OPM graphs are depicted using the following conventions: *(i)* circles are artifacts, rectangles are processes, and octagons are agents; *(ii)* an edge from an artifact to a process is of type `opm:wasGeneratedBy`, from process to artifact is `opm:used`, from artifact to artifact is `opm:wasDerivedFrom`, from process to agent is `opm:wasControlledBy`; *(iii)* we omit these types from the figures for brevity, except if a role is specified, where the role is in brackets after the edge type as in `used(R)` — if not specified, it has the value `opm:undefined`, to be replaced in application-specific mappings with informative roles; *(iv)* an annotation to an edge is a label of the form 'X=Y', where X is the annotation key and Y is the value; *(v)* a timestamp on an edge is written 'T'.

We employ a simple illustrative example from an existing work on DC [6]: *A book entitled "The Library as Literacy Classroom" was authored by Marguerite C. Weibel and published by the American Library Association (ALA) in 1992.* We only provide mappings for provenance-related DC terms, i.e. those which assert something about the past.

First, we map between assertions of `dc:creator` and the corresponding OPM pattern. The creation of a resource is a process which produces the generation of that resource for which there is no prior generation. The creator is an OPM agent with the creator's name (`opm:pname` annotation). As in DC, `opm:pname` can take unique identifiers following different naming schemes.

Applicable Dublic Core Expressions	
R *creator* C	An entity primarily responsible for making R.

OPM Pattern



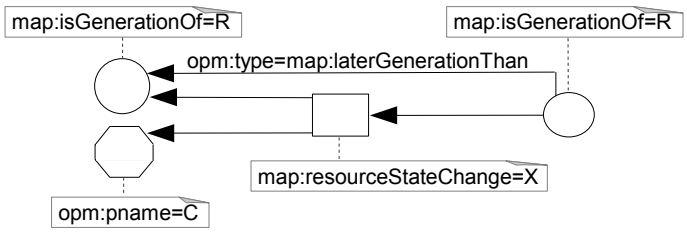
Constraints. There is no other artifact for which the artifact in the pattern links to with `map:laterGenerationOf`.
Description. A process occurred that created the first generation of resource R (artifact), and this was performed by entity C (agent).
Example. An authorship process occurred that created the first generation of "The Library as Literary Classroom" (R), and this was performed by Marguerite C. Weibel (C).

Creator Mapping

Next, an assertion of `dc:contributor` or `dc:publisher` can be mapped as follows. Contributing/publishing is a process with the effect of making the resource modified/available. We annotate this process in the graph with a `map:resourceStateChange` annotation, asserting that the process is of a kind which changed the state of a resource in the named manner.

Applicable Dublic Core Expressions	
R *contributor* C	An entity responsible for making contributions to R, *or*
R *publisher* C	An entity responsible for making R available

OPM Pattern



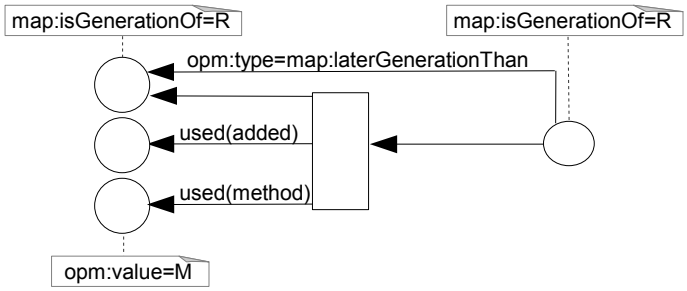
Constraints. X=available or X=modified, as appropriate.
Description. A process occurred, performed by entity C, that changed resource R to a new generation, so it was afterwards available/modified.
Example. A publishing process occurred that changed "The Library as Literary Classroom" (R) from one generation to another, such that it was afterwards available, and this was performed by the ALA.

Affector Mapping

A `dc:accrualMethod` assertion states the general accrual method for a collection, e.g. Deposit, Donation, Purchase [1], not a process' execution. The OPM graph depicts an addition to the collection, with artifacts for method and data added to the collection. We use `opm:value` to assert the method's value, but if a unique reference is given, `opm:pname` may be appropriate.

Applicable Dublic Core Expressions	
R *accrualMethod* M	The method by which an item was added to R.

OPM Pattern



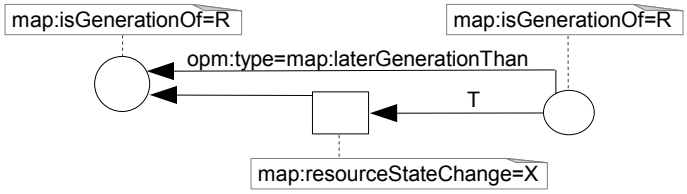
Description. A process occurred, applying method M, taking one generation of collection R and added an item, creating a new generation.
Example. A process occurred, applying LoC's archival method, taking one generation of the library (R) and added an item to it ("The Library as Literary Classroom"), creating a new generation of the library.

Accrual Method Mapping

The following maps DC assertions stating dates to OPM timestamps.

Applicable Dublic Core Expressions	
R *available* T	Date that R became available, *or*
R *dateAccepted* T	Date of acceptance of R, *or*
R *dateCopyrighted* T	Date of copyright, *or*
R *dateSubmitted* T	Date of submission of R, *or*
R *modified* T	Date on which R was changed, *or*
R *valid* T	Date of validity of R.

OPM Pattern



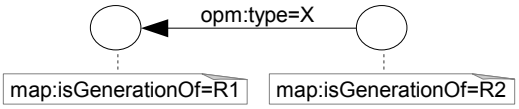
Constraints. X=available, accepted, copyrighted, submitted, modified, or valid, as appropriate.
Description. A process occurred that changed the context or value of resource R from one generation to another, such that it was afterwards X (available/accepted/...), and this happened at time T.
Example. A publishing process occurred that changed the "The Library as Literary Classroom" (R) from one generation, unavailable, to another, such that it was afterwards available, and this happened in 1992 (T).

Date Mapping

Finally, some DC assertions relate (generations of) resources. In reading the pattern below, it is important to remember we are mapping only *provenance* information from DC assertions (how something came to be as it is), else the result can seem counter-intuitive. For example, the assertion "California is part of the US" does not obviously entail the OPM interpretation "the US is derived from California" as implied by the mapping. However, if rephrased "How did the US come to be as it is?", then we may reasonably assert a contributing reason "California is part of the US", i.e. what the US *is* derives, in part, from what California *is*.

Applicable Dublin Core Expressions	
R1 $isPartOf$ R2	A related resource in which R was physically or logically included, *or*
R2 $hasPart$ R1	A related resource that was included physically or logically in R, *or*
R1 $isVersionOf$ R2	A related resource of which R was a version, edition or adaptation, *or*
R2 $hasVersion$ R1	A related resource that is a version, edition or adaptation of R, *or*
R1 $isReplacedBy$ R2	A related resource that supplanted, displaced, or superseded R, *or*
R2 $replaces$ R1	A related resource that was supplanted, displaced, or superseded by R, *or*
R1 $isReferencedBy$ R2	A related resource that referenced, cited, or otherwise pointed to R, *or*
R2 $references$ R1	A related resource that was referenced, cited, or otherwise pointed to by R, *or*
R2 *source* R1	A related resource from which R is derived.

OPM Pattern



Constraints. X=contained for $hasPart/isPartOf$, X=hadVersion for $hasVersion/isVersionOf$, X=replaced for $replaces/isReplacedBy$, X=referenced for $references/isReferencedBy$. No annotation is present for *source*: it is simply a `opm:wasDerivedFrom` relationship.
Description. Two resources were related so that one depended on the other's existence.
Example. The article "Mapping Attribution Metadata to the Open Provenance Model" (R2) was related to the book "The Library as Literary Classroom" (R1) by referring to it, such that the former's existence depended on the latter's.

Interrelation Mapping

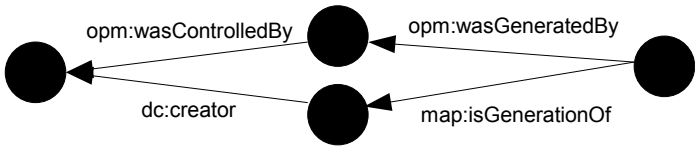


Figure 1: An RDF graph for dc:creator with combined OPM and DC information

3.3. Mapping and RDF

In some applications, a set of DC metadata could be translated to an OPM graph expressed in RDF, and therefore as a subset of an RDF graph. Comparably, DC terms may be translated from OPM attribution patterns expressed as a subset of an RDF graph. To map from a pattern found in an OPM graph to DC RDF, we need to search for that pattern and then construct the corresponding DC triple. For example, a SPARQL query could take an RDF serialisation of an OPM pattern and construct a DC relation between entities. Assuming a trivial RDF serialisation of OPM, where one edge maps to one RDF triple, we could write the following query to construct `dc:contributor` assertions where an OPM graph expresses a contribution process by a known agent.

```
CONSTRUCT { ?r dc:contributor ?c }
WHERE {?a2 map:laterGenerationThan ?a1. ?a2 opm:wasGeneratedBy ?p.
       ?p opm:wasControlledBy ?c.       ?p opm:used ?a1.
       ?a1 map:isGenerationOf ?r.       ?a2 map:isGenerationOf ?r}
```

This would not be an expressive enough serialisation for generic OPM, as we could not include roles, timestamps or other edge annotations, and the official OPM OWL serialisation [2] instead expresses OPM graph edges as RDF resource instances. A comparable query using that ontology would expand to include RDF triples matching assertions connecting OPM nodes and edges.

In both directions of translation, it is possible to result in both the original data and its translation in one RDF graph. Figure 1 shows an adaptation of the pattern shown for `creator` above, with OPM edges and annotations mapped to RDF edges, with the addition of the DC relation between resource and creating user.

3.4. Unstructured Provenance Assertions

Two DC terms refer to general expressions of provenance information, without specifying what structure it takes, and so cannot be mapped to OPM graph structures. A *bibliographic citation* is defined as "A bibliographic reference for the resource" [1]. As bibliographic information may be contained in an OPM graph (the process of creation by the authors, the process of publication by the journal etc.), the bibliographic citation can be seen as the results of a query over the OPM graph.

Comparably, *provenance* is defined as "A statement of any changes in ownership and custody of the resource since its creation that are significant for its

authenticity, integrity, and interpretation." [1]. In OPM, the provenance of an artifact is an OPM graph in which there is a path from that artifact to every node (process, artifact or agent). DC-style provenance concerns the history of an artifact at a particular level of granularity, including only particular types of process, i.e. processes related to transfer of ownership. These are comparable concepts, but remain too loosely specified for a concrete mapping to be specified.

4. Usage

We see the greatest benefits coming from playing to the strengths of each of the causal graph and attribution metadata approaches to provenance. Specifically, we see OPM being used in communication between software components, and DC being used in communication between users and software. Automatic mapping would then exist between the two, with data in both models co-existing where communication with both users and software is valuable. This approach would allow the simplicity, speed and intuitiveness of attribution metadata approaches, and the expressiveness and querying advantages of causal graph models. DC already has proven experience in use as the basis for intuitive user interfaces for inputting attribution metadata, e.g. [7].

Consider the following illustrative use case. A repository contains a text resource R marked up with '`dc:contributor` C' and '`dc:dateSubmitted` D'. Archivist A is familiar with OPM and wants to provide more information about the processes that the metadata implies: that the contribution was by editing in MS Word and this happened (at an unknown time) prior to submission. The OPM-DC mapping is applied to create OPM graphs from the DC statements. The `dc:contributor` mapping result contains artifacts for the resource before and after editing, an agent C and a process which U annotates to show it denotes MS Word editing (see Affector Mapping in previous section). The `dc:dateSubmitted` mapping result contains artifacts for the resource before and after submission, and a process denoting the submission (see Date Mapping). A connects the two graphs by asserting that the 'before' artifact of the `dc:dateSubmitted` graph derives from the 'after' artifact of the `dc:contributor` graph. The OPM graph is added to the repository, with the 'isGenerationOf' annotations linking the artifacts in the graph to the original resource R. Later, A discovers that R, in its state prior to the documented contribution, was referenced by another resource S, and so adds a `opm:wasDerivedFrom` edge of type `referenced` linking to the appropriate artifact (generation of R) with an artifact labelled as being a generation of S. User V, unfamiliar with OPM, then wishes to browse information about R and downloads the OPM graph. Before displaying this metadata to V, the graph is mapped back to DC and presented as three statements: '`dc:contributor` C' , '`dc:dateSubmitted` D' and '`dc:isReferencedBy` S' (see Interrelation Mapping).

5. Related Work

As an alternative to mapping existing models, some approaches combine causal graphs and attribution metadata, by providing vocabulary for expressing DC-like assertions while also asserting causal relationships, and so exhibit the strengths and the weaknesses of both. Such approaches include: *(i)* the Provenir ontology [8], with DC-like vocabulary including `part_of` and `has_temporal_value` plus interconnections between data, processes and agents to express multiple steps of derivation over time; *(ii)* the Provenance Vocabulary Core ontology [9] which includes many of the same concepts as DC, but is specific to provenance and focuses on describing the processes which produced the data resources; *(iii)* the Proof Markup Language (PML) [10], which again includes vocabulary for both expressing attribution and process, focusing on inference processes with the provenance information expressing how some result was inferred through a series of rules from initial data; and, *(iv)* OAI-ORE [11], which concentrates on describing aggregations of resources, including terms comparable to DC for expressing collection-part relationships, *aggregates* but also statements of where a resource is sourced from an aggregation (*lineage*). However, in many of these cases, resources are, as in DC, mutable, leading to potential ambiguity over which generation of a resource an assertion refers.

PREMIS [12] is perhaps a closer match to OPM, with *representations* being particular serialisations of preserved information (*intellectual entities*) and therefore apparently immutable data items. These can relate to each other via *events*, for example denoting the difference between two representations due to a conversion operation or a change in metadata. PREMIS metadata focuses on preservation matters, so primarily concerns access and interpretation of archived objects.

6. Conclusions

Mapping from Dublin Core to the Open Provenance Model brings potential benefits for users of both models. First, by giving explicit representations for acts of creation, contribution and publication, and the intermediate versions leading up to the final collection, we have a hook on which to provide additional information about those actions and versions, i.e. it is clear what extra information is needed for a comprehensive description of what occurred. Second, we are now able to connect this metadata with other descriptions in OPM, such as documentation of the archival process for a collection, or more detailed steps of the process by which the collection was created, i.e. the attributed events become queryable as part of a wider history. Third, we have two representations for communicating the same metadata: it can be interpreted not only those services which understand DC, but also those which can parse OPM. By being able to be manipulated by more tools, we can get more value from the data. Finally, we have a way to reduce particular patterns in OPM graphs to more user-friendly explanations of attribution, where the requisite mapping-specific annotations are present.

Acknowledgements: We are grateful for much helpful feedback on earlier versions of the mapping and pointers to material from Jim Myers, Joe Futrelle, Thomas Habing, Luc Moreau, Paolo Missier, and this paper's reviewers.

References

[1] Dublin Core Metadata Initiative, DCMI Metadata Terms, http://dublincore.org/documents/dcmi-terms/ (December 2009).

[2] L. Moreau, B. Clifford, J. Freire, Y. Gil, P. Groth, J. Futrelle, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, Y. Simmhan, E. Stephan, J. V. den Bussche, The open provenance model core specification (v1.1), Future Generation Computer Systems (this volume).

[3] D. G. Campbell, The use of the Dublin Core in web annotation programs, in: International Conference on Dublin Core and Metadata Applications, Dublin Core Metadata Initiative, Florence, Italy, 2002, pp. 105–110.

[4] T. Baker, A Grammar of Dublin Core, D-Lib Magazine 6.

[5] METS, Metada Encoding and Transmission Standard (METS), http://www.loc.gov/standards/mets/ (December 2009).

[6] S. L. Weibel, The state of the Dublin Core Metadata Initiative April 1999, D-Lib Magazine 5 (4) (1999) Online publication.

[7] M. N. Boulos, A. V. Roudsari, E. R. Carson, Towards a semantic medical Web: HealthCyberMap's tool for building an RDF metadata base of health information resources based on the Qualified Dublin Core Metadata Set., Medical Science Monitor 8 (2002) 124–36.

[8] S. S. Sahoo, R. S. Barga, J. Goldstein, A. P. Sheth, K. Thirunarayan, Where did you come from... Where did you go?, Tech. rep., Kno.e.sis Center, Wright State University, Dayton, US (2009).

[9] O. Hartig, J. Zhao, Provenance Vocabulary Core Ontology, http://purl.org/net/provenance/ns (December 2009).

[10] P. Pinheiro da Silva, D. L. McGuinness, R. Fikes, A proof markup language for semantic web services, Information Systems 31 (4-5) (2006) 381–395.

[11] C. Logoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, S. Warner, Open archives initiative object reuse and exchange, http://www.openarchives.org/ore/vocabulary (October 2008).

[12] PREMIS Editorial Committee, PREMIS Data Dictionary for Preservation Metadata, version 2.0, http://www.loc.gov/standards/premis/ (March 2008).